

An Innovative Method to approach Operational Feasibility Study for Making the Software Popular

Prof Rasendu Mishra, Assistant Professor,
MCA Section, Department of Computer Engineering,
Institute of Technology, Nirma University, Ahmedabad
Gujarat, India
Email: rasendu1@gmail.com

Prof Rushabh Shah, Assistant Professor,
MCA Section, Department of Computer Engineering,
Institute of Technology, Nirma University, Ahmedabad
Gujarat, India
Email: rushi_nirma@yahoo.co.in

Prof Ajaykumar M Patel, Assistant Professor,
Department of Computer Engineering,
Institute of Technology, Nirma University, Ahmedabad
Gujarat, India
Email: ajaypatel@nirmauni.ac.in

Abstract: Software has influenced every one of our lives in a considerable manner. Software development and software success is yet an indistinguishable test from it was beforehand. This paper is recommending the most vital software feasibility test to be performed before going to other traditional and expressed software feasibility tests. The thought is to reinforce the recommended software feasibility test in order to expand the success of the software.

Keywords: *Feasibility study, software success, software development*

Introduction: In today's world computer science has touched every aspect of the lives of the human race. Today we have software for generating bills at malls, for controlling and running small-mid-sized-large-vary-large businesses, for medical help, for security purpose of a country, for teaching-learning-research purpose, for interpreting and concluding data and for many more purposes. Some of these software are custom built, some are built by outsourcing the development work or by using off the self-components and the current buzz is using Cloud based software solutions so the problem of scale up and scale down and cost can be reduced to a larger extent.

The software is developed, by any of the above stated means, using a specific approach of Software Development Life Cycle and a specific software development model. But there is a basic question "Are all software, so developed and deployed, successful"? "Are all software, so developed, gives the tented results to the companies for which it was developed"? "Are the users of the software using full functionalities so provided by the software or in other words is the software used to the best of its functional limits?"

There are many more such questions related to the successful deployment, usage and acceptance of the software amongst those for which it was intended for. And if the answer to the questions are negative then the rate-of-return from the software will not be as intended. In whole, the software is not a "Popular Software".

Problem Statement: The problem which is stated in this paper is:

1. How to make a software popular?
2. How to increase its acceptance level?
3. How to increase trust of the users on the software?
4. How to strengthen the users so that they would use all the functionalities of the software, whether simple or complex, without any

hesitation leading to increase the software usage for all walk of their use?

5. How to create a situation where users themselves represent the need for the software? A problem highlighted in Operational Feasibility.

Reasons:

A large text appears in books, online websites, and magazines giving a variety of reasons for non-acceptance of above goals. Some of them are mentioned below:

- Improper Requirement Engineering.
- Improper Analysis and Design of the software
- Improper Timelines and Budgets
- Poorly designed GUI.
- Poor Communication while development
- Poor testing, deployment and training

Previous Solutions:

A large number of solutions appears stating that due to above reasons a large number of software fails or do not get the success ratio as intended leading to less of rate of return on investment (RoI). Solutions have been proposed like Agile Software development, Modular and Object Oriented Software Development, Using Software for Project Management to determine realistic timelines, Human Resource availability and budget requirements. Even though the software utility is not much increased.

Proposed Solution:

This paper proposes a solution to make a deployed software a success. This can be done by enhancing operational feasibility study which is to be partly carried out before other feasibility checks.

“User Attitude Feasibility Check” should be done before other feasibility checks are done. In this, a user should be asked to perform a particular complex task in the current usual way. The software developers should prepare a demo software to perform that same task. Now when the user returns with the completed task show the user how fast and accurately it can be done, if done using a software rather manually. This conversation should take place in the most polite manner. The objective of this work is to involve the users actually, at grass root level, to know the benefit of the software so developed.

How It works with scenario:

1. The system analyst should ask the employee to collect and analyse the data of sales of products of last 3 years.
2. The system analyst should keep on other hand a demo software ready to generate the similar results.
3. Its obvious the employee would take 4 to 5 hours or more to solve above problem.
4. When employee comes with the solution, the system analyst should thank and praise him but very politely generate the same solution using software in fraction of time and thus explain the use of software in saving the time and efforts and generating the solution with full accuracy.
5. The system analyst should ask the employees to generate few more results from the demo software and thereby building a positive attitude towards the software.

Advantages of the proposed solution:

1. User will be given a demo showing how their work can be done with an ease and efficiency.
2. Users will get motivated and fear to use software would be reduced.
3. Users will be motivated towards software usage.
4. More involvement of users, of grass root level, would increase and they would start giving suggestions in software development. Hence communication between users and developers would be strengthened.
5. The developed software would be utilized to its maximum efficiency leading higher RoI.

-
- Limitations:**
6. A user friendly and user liked software would be developed.
 7. Training time may be reduced.
 1. This will be more suitable for companies opting for manual to computerized software solution for the first time.
 2. Time to develop software may increase due to more people involvement during requirement engineering phase.
 3. Demo software, which might be quality, may be extended towards final software modules.
 4. It is applicable for a successfully developed and deployed software.

Conclusion: In order to make a software more popular and in order to increase the utilization of the software to its maximum efficiencies, we need to bring its highest acceptance amongst users of the software. This may be achieved if we involve the users of all levels and show them a comparative demo before software development and try to remove fear towards the software usages then this would increase the trust of the users on the software and hence the failure rate of the deployed software would decrease.

- References:**
1. Kendall & Kendall, “System Analysis and Design”, Chapter 3, Edition 8.
 2. Roger Pressman, “Software Engineering”, Latest Edition.